

Recovery of Sharp Features in Mesh Models

Zhao Liu¹ · Maodong Pan¹ · Zhouwang Yang¹ ·
Jiansong Deng¹

Received: 7 February 2015 / Revised: 9 April 2015 / Accepted: 15 April 2015 /

Published online: 5 June 2015

© School of Mathematical Sciences, University of Science and Technology of China and Springer-Verlag Berlin Heidelberg 2015

Abstract Due to the shortages of current methods for the recovery of sharp features of mesh models with holes, this paper presents two novel algorithms for the recovery of features (especially sharp features) in mesh models. One algorithm defines an energy that is regarded as the difference between the initial features and the ideal features. The optimal solution of the energy optimization problem modifies the initial features. The algorithm has good performance on sharp features. The other method establishes a plane cluster for each initial feature point to obtain a corresponding modified feature point. If necessary, we can obtain the modified feature line by fitting these modified points. Both methods depend little on the result of filling model holes and result in better features, which maintain the sharp geometric characteristic and the smoothness of the model. The experimental results of the two algorithms demonstrate their superiority and rationality compared with the existing methods.

Keywords Hole repair · Sharp feature · Mesh models

Mathematics Subject Classification 65D17

1 Introduction

With the widespread application of reverse engineering, scanning and reconstruction of 3-dimensional entities is being applied more and more widely. 3-dimensional entity scanning reconstruction scans objects in different directions and then synthesizes digital point cloud models by multiple overlaying of these discrete point clouds of the

✉ Jiansong Deng
dengjs@ustc.edu.cn

¹ School of Mathematical Sciences, University of Science and Technology of China, Hefei, China

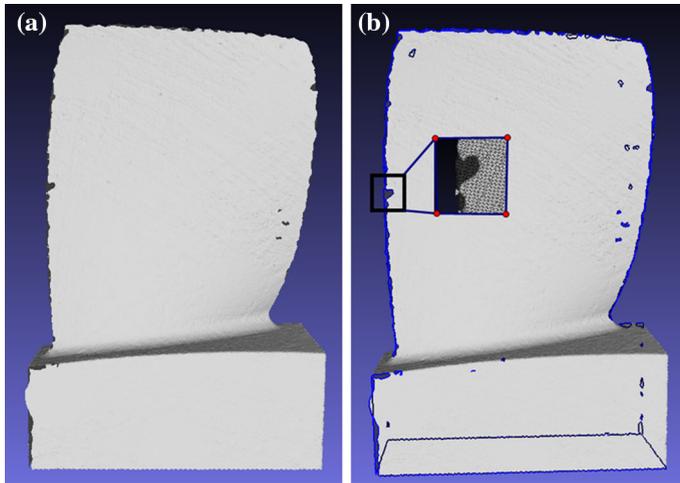


Fig. 1 The real model of scanning an entity

sampling surface. However, the result of this method usually results in a loss of the sharp features of the initial model and sometimes causes holes that will result in the loss of critical information and causes difficulties in model reconstruction.

For example, when scanning entities with thin edge parts, it is difficult to accurately obtain the thin edge parts. Usually, the thin edge parts are lost in the scanning model. We encountered a similar problem in practice when scanning an entity (see Fig. 1). Figure 1a shows the raw model of the scanning entity. The blue parts in Fig. 1b indicate the holes of the scanning model. From the result, we can see that the thin edge parts cause holes in the scanning entity, which lead to the loss of features.

Features are critical compositions of a model that can reflect the characteristics and geometric information. They are important parts of geometric modeling and model reconstruction in computer graphics, CAD, and engineering. The lack of sharp features affects the correction of model visualization, model reconstruction and some post-processing, such as isogeometric analysis and finite element mesh analysis. In most situations, there are noises and holes in scanning model. Therefore, it is necessary to denoise the scanning model and repair the holes before restoring the sharp features.

The existing methods in repairing models can repair model holes in meshes but cannot meet expectations in handling sharp features. Some methods just fill holes without consideration of the restoration of the feature; some methods attempt to fill holes on the premise of maintaining sharp features or recovering sharp features on repaired models [3, 13, 17, 21], and others try to restore sharp features on initial model [3, 7]. Unfortunately, the effects are not obvious. In fact, these repair methods tend to change the sharp features into smooth lines while repairing the original model and maintaining global smoothness. It is a significant obstacle in model reconstruction.

Aiming at the shortage characteristics of the recovery mesh model methods, this paper proposes two algorithms to recover sharp features: one algorithm is based on energy and minimizes the L1-norm energy to obtain the theoretical feature line of scanning models; the other is based on geometric characteristics to establish plane

clusters and uses the adjacent points' positions and the normals of the feature points to modify every initial point and then fit them to generate a good result. They perform better than former methods experimentally.

2 Previous Work

The work proposed in this paper relates to previous researches conducted in the areas of: denoising, mesh model repair, sharp feature recovery techniques, which are consecutively reviewed below.

2.1 Mesh Model Denoising and Repair

There are varieties of surface denoising algorithms and the most common techniques are based on Laplacian operators [11, 27]. Fleishman et al. [13] and Jones et al. [17] applied the bilateral filter on mesh denoising and [26, 32] improved the method. Wang et al. proposed a method for denoising with a discrete Laplacian regularization smoothing model and the resulting denoised mesh was proved asymptotically optimal [29]. In this paper, the method in [29] is used for denoising the scanning model before sharp feature recovery.

Previously, repairing holes in mesh models usually involved meshing holes directly or adding some appropriate vertices to the model and then grid holes. Barequet et al. triangulated holes directly to fill gaps [5, 6]. Davis et al. and Fakir and Turk used volumetric techniques to repair models [10, 24]. Liepa first triangulated holes and then placed new vertices within the newly inserted triangles' new vertices to produce the desired vertex density and finally moved new vertices to positions that minimized an appropriate function and made the patch's normal field fairly continuous with respect to the neighboring surface [22]. However, [6, 10, 22] cannot handle cases of sharp features, which result in losing them. Sharf et al. filled holes iteratively by copying patches from valid regions of the given surface [25]. Chen et al. used the radial basis function to approximate the missing meshes to generate triangles [8]. Unfortunately, these methods cannot restore sharp feature lines/angles correctly.

2.2 Sharp Feature Recovery

During the past years, a wide variety of sharp feature recovery methods have been proposed. Kobbelt et al. proposed a feature preserving approach to surface extraction from volume data [20]. Ju et al. improved Kobbelt's method [18], and the improved method was used for mesh repair with feature preservation [19]. These approaches depend on an accurate detection of feature regions. Fleishman et al. presented a Moving Least Square approach to reproduce sharp features [12] and then, Daniels et al. built up on [12] to project points onto the features of the data set [9]. Lipman et al. also presented an MLS fitting method to handle sharp features [23] but it needs careful fine tuning in the presence of noise and has problems with structures like the corner of a cube.

Avron et al. reconstructed surfaces from point clouds while respecting sharp features [4], but the method can produce less than satisfactory results, for example, erroneous

normal estimates, near sharp features as a result of severe noise and undersampling. Huang et al. proposed an algorithm to produce consolidated point sets with noise-free normals and clean preservation of sharp features [15]. The method improved the performance of edge-aware reconstruction methods and point set rendering techniques but it requires dense points around sharp features, which is not always available in a scanned point cloud data.

For sharp feature recovery in triangle meshes, Hubeli et al. proposed a method for extracting a multi-resolution organization of sharp edges from a triangle mesh in [16] but the process may become slow for dense meshes with many sharp features. Biermann et al. subdivided the mesh in order to obtain sharp features in the limit surface [7]. Hoppe et al. obtained higher-resolution features by inverting the edge contractions through vertex splits [14]. The identification of perceptually salient curvature extremum was used to detect curvature features in [31]. Both [14, 31] may result in the identification of a set of lines corresponding to small radius blends in the input model.

A hole filling with feature preserving method was stated by Wang et al. [30]. The approach can automatically recover the missing sharp features and corners from piecewise smooth surfaces. However, a sequence of processes including hole detection, feature detection, feature classification, feature sets matching, convex/concave analysis, and further checking are needed before curve reconstructing. The preprocessing is time-consuming and complicated. After preprocessing, the recovery method of feature line is just fitting the matching feature points, which couldn't recover the missing sharp feature sufficiently. An energy method based on the volume of tetrahedron was also proposed to recover the corner points. In our paper, we define a different energy and mainly use it to recover sharp feature lines.

Attene et al. proposed a method to sharpen sharp edges on filled holes mesh models by inserting new vertices to generate sharp features in a small range neighborhood, which may possess sharp features [1–3]. To some extent, the sharpening processing methods can recover the lost features. However, there are still two obvious shortcomings: first, the effect of feature recovery depends on the result of repairing holes. The sharpening process only modifies the features in a small range. If the repaired result feature is far from the actual feature, the sharpening process is not as useful. Second, the method inserts new vertices locally without consideration of the model's global smoothness and other geometric characteristics. This will lead to a rough model with irregular edges and will have a bad influence on entity reconstruction.

Taking into account the advantages and disadvantages of the existing methods, this paper proposes two methods to restore sharp features on the filled holes model. The methods depend minimally on the result of repairing meshes and greatly enhance the accuracy. Therefore, they have better behavior for sharp feature recovery than former methods.

3 Contribution

Two sharp feature recovery algorithms are proposed in this paper. One is based on energy optimization and the other is based on plane cluster. Compared with the existing approaches for similar purposes, our methods show the following advantages:

1. A novel algorithm involved L1-norm and quadratic form is approached to restore sharp features, in which, sharp feature recovery problem is transformed into an energy optimization problem. With the help of the convenient methods for solving the convex optimization problem in numerical optimization research, our algorithm is robust and efficient. The algorithm restores the sharp feature and keeps the smoothness of the feature at the same time.
2. Another algorithm based on plane cluster approached in this context modifies features of the given model point by point instead of modifying the feature once globally. It has a good performance on sharp feature recovery, especially in the recovery of sharp corner points.
3. When restoring the sharp features, our algorithms only use points of initial model (excluding added points for model repair). Therefore, though both algorithms are on the premise of mesh model repair, they depend little on the result of the filling holes features. The probable position of the filling holes features can be used to recover the ideal sharp feature instead of sharpening sharp edges in repaired model, which depends a lot on the result of filling holes.
4. Besides, our approaches only restore the meshes around sharp features and maintaining the other parts of the given model which will prevent the shrinkage and deformation of the non-sharp region.

4 Sharp Feature Recovery Based on the Energy Optimization Algorithm

Initial model information, the most credible data collected from the original physical model, is very important in the mesh model reconstruction process, with no exception in restoring sharp features.

In breakage models, sharp features are often located at the intersection line of natural extension of the two surfaces, see Fig. 2. Taking this point into consideration, this paper puts forward two methods to restore sharp features on the basis of filling holes. For simplicity, the features of the repaired model are called “the repaired features”. We use the repaired features to approximate ideal features. The filling holes method is used according to [1] to repair holes to obtain a complete grid model, which consists of two parts: initial mesh grid points and vertices added to fill holes. In recovery processing, we trust initial point information more than the added points.

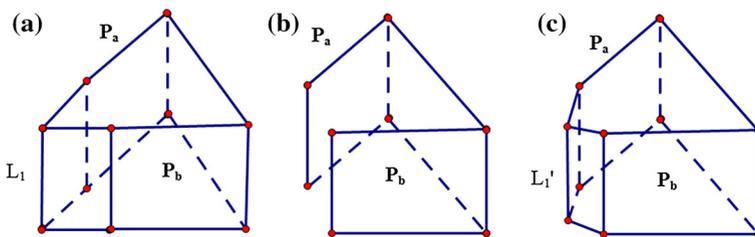


Fig. 2 Ideal feature and repaired feature. In **a** the sharp feature L_1 is located at the intersection of natural extension of planes P_a and P_b . **b**, **c** indicate the breakage model and the *filling holes* result with its sharp feature line L'_1 . The features of the *filled holes* model are “repaired features” and they are usually different from the ideal feature

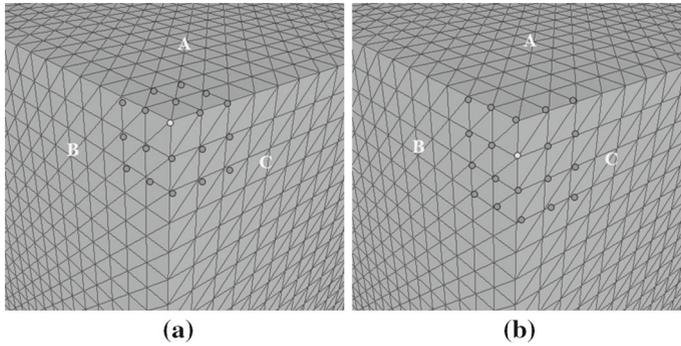


Fig. 3 Determine the adjacent points of the *light points*. In **a** there are three surfaces related to the *light point*. The adjacent points should be chosen almost equally from surfaces A, B, and C. In **b** there are two surfaces related to the *light point* and the adjacent points should be chosen almost equally from surfaces B, C

In most situations, the features of the model satisfy some smoothness. Assume that the sharp feature line of the general model can be indicated by a p th-degree B-spline curve $P(t)$:

$$P(t) = \sum_{k=0}^n \mathbf{P}_k N_{k,p}(t), \tag{4.1}$$

where $\{\mathbf{P}_k\}_{k=0}^n$ are the control points of $P(t)$ and $\{N_{k,p}(t)\}$ are the p th-degree B-spline basis functions.

Suppose the repaired feature points are $\{\mathbf{Q}_i\}_{i=0}^m$. Dihedral method is used to detect $\{\mathbf{Q}_i\}$ and the user can select or delete some points interactively. Parameterize $\{\mathbf{Q}_i\}_{i=0}^m$ and approximate these feature points $\{\mathbf{Q}_i\}_{i=0}^m$ with a p th-degree nonrational B-spline curve $\tilde{P}(t) = \sum_{k=0}^n \tilde{\mathbf{P}}_k N_{k,p}(t)$. t_i is the parameter value of each \mathbf{Q}_i using the parameterization method, such as the chord length method and so on.

For each feature point \mathbf{Q}_i , determine the adjacent points $\{\mathbf{q}_{ij}\}_{j=0}^N$ and make $\{\mathbf{q}_{ij}\}_{j=0}^N$ almost equally distributed on each related surface (see Fig. 3). The adjacent points mean the most nearest neighbor points to \mathbf{Q}_i on the model and N denotes the quantity of them. We use k -nearest neighbors to determine q_{ij} . For example, if \mathbf{Q}_i has 3 related surface A, B and C and we want to get 30 adjacent points of \mathbf{Q}_i . Denote S_A, S_B and S_C as the corresponding adjacent point sets on surface A, B and C. First, we find a little more than 30 (such as 35 or 40) adjacent points $\{q_{ij}\}$ by k -nearest neighbors in the order of distance. Second, pick 10 points from $\{q_{ij}\}$ which are located on A to join S_A . It is easy to classify them by normals. Then, do the same to S_B and S_C . Finally, the points in S_A, S_B and S_C are the adjacent points of \mathbf{Q}_i . One may worry that the size of S_A, S_B or S_C could be small than 10. If it happens, we just enlarge the size of adjacent set in the beginning. In particular the size of the neighborhood strongly affects the result and therefore the size can be set by the user interactively to get the best result. As the points on the initial model are more reliable than the points added to fill holes, we should abandon added points and 1-neighbor points (bad normals)

when choosing adjacent points. The examples for choosing adjacent points are in the following parts (see Figs. 6b, 8b).

As the ideal feature line $P(t)$ can be considered as the modification of the repaired feature line $\tilde{P}(t)$, we treat $\{\mathbf{q}_{ij}\}_{j=0}^N$ as the adjacent points of ideal feature point $P(t_i) = \sum_{k=0}^n \mathbf{P}_k N_{k,p}(t_i)$, $i = 0, \dots, m$.

Denote

$$E_{\text{inter}} = \sum_{i=0}^m \sum_{j=0}^N |(P(t_i) - \mathbf{q}_{ij})^T \mathbf{n}_{ij}|, \tag{4.2}$$

where t_i is the parameter of \mathbf{Q}_i and \mathbf{n}_{ij} is the normal of the adjacent point \mathbf{q}_{ij} .

Equation 4.2 defines an energy E that reflects the difference between the B-spline curve $P(t)$ and the intersection line of the two surfaces. Minimize the energy E to obtain the control points of B-spline curve $P(t)$. Taking the smoothness of $P(t)$ into consideration, we add a modification term E_{smooth} to E_{inter} and get an energy E (see Eq. 4.3).

$$E_{\text{smooth}} = \mu \int_0^1 |P''(t)|^2 dt, \tag{4.3}$$

$$E = E_{\text{inter}} + E_{\text{smooth}},$$

where μ is a constant term that can be set manually. Solve the optimization problem $\arg \min E$ to obtain the control points of $P(t)$ and the $P(t)$ is regarded as the modification of repaired feature line $\tilde{P}(t)$. The specific steps of the method are as follows:

1. Fill holes of the initial model and find the repaired feature points $\{\mathbf{Q}_i\}_{i=0}^m$.
2. For each \mathbf{Q}_i , determine the adjacent points $\{\mathbf{q}_{ij}\}_{j=0}^N$ and try to make $\{\mathbf{q}_{ij}\}_{j=0}^N$ almost equally distributed on the surfaces on both the sides (see Fig. 3).
3. Define energy E as :

$$E = \sum_{i=0}^m \sum_{j=0}^N |(P(t_i) - \mathbf{q}_{ij})^T \mathbf{n}_{ij}| + \mu \int_0^1 |P''(t)|^2 dt.$$

4. The best estimation of the original sharp features can be obtained by solving $\mathbf{P} = \arg \min E$, where \mathbf{P} is denoted as the control points of $P(t)$.

As seen from the definition of the energy E , the first term reflects the difference between the target curve and the intersection of surfaces; the second term indicates the smoothness of the target curve. Optimizing the energy to minimize E , the optimization solution is the best estimation of the feature. The method automatically includes modifying the feature points to the ideal position and then fitting them globally.

To solve $\mathbf{P} = \arg \min E$, Eq. 4.1 demonstrates that:

$$\int_0^1 |P''(t)|^2 dt = \int_0^1 \left| \sum_{k=0}^n \mathbf{P}_k N''_{k,p}(t) \right|^2 dt = \mathbf{x}^T \mathbf{Kx}, \tag{4.4}$$

where $\mathbf{x} = (\mathbf{P}_0^T, \mathbf{P}_1^T, \dots, \mathbf{P}_n^T)^T$ denotes the control-point vectors . The matrix \mathbf{K} is defined as follows:

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{00} & \mathbf{K}_{01} & \cdots & \mathbf{K}_{0n} \\ \mathbf{K}_{10} & \mathbf{K}_{11} & \cdots & \mathbf{K}_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{K}_{n0} & \mathbf{K}_{n1} & \cdots & \mathbf{K}_{nn} \end{pmatrix} \tag{4.5}$$

in which, $\mathbf{K}_{ij} = \lambda_{ij} \mathbf{I}_3$, $\lambda_{ij} = \int_0^1 N''_{i,p}(t)N''_{j,p}(t) dt$. Therefore,

$$\min E = \min \left(\sum_{i=0}^m \sum_{j=0}^N |(P(t_i) - \mathbf{q}_{ij})^T \mathbf{n}_{ij}| + \mu \mathbf{x}^T \mathbf{K} \mathbf{x} \right). \tag{4.6}$$

Moreover,

$$\sum_{i=0}^m \sum_{j=0}^N |(P(t_i) - \mathbf{q}_{ij})^T \mathbf{n}_{ij}| = \sum_{i=0}^m \sum_{j=0}^N \left| \left(\sum_{k=0}^n \mathbf{P}_k N_{k,p}(t_i) - \mathbf{q}_{ij} \right)^T \mathbf{n}_{ij} \right| = \|\mathbf{A} \mathbf{x} - \mathbf{b}\|_1 \tag{4.7}$$

in which,

$$\mathbf{A} = \begin{pmatrix} N_{0,p}(t_0) \mathbf{n}_{00}^T & \cdots & N_{n,p}(t_0) \mathbf{n}_{00}^T \\ \vdots & \cdots & \vdots \\ N_{0,p}(t_0) \mathbf{n}_{0N}^T & \cdots & N_{n,p}(t_0) \mathbf{n}_{0N}^T \\ \vdots & \cdots & \vdots \\ N_{0,p}(t_m) \mathbf{n}_{m0}^T & \cdots & N_{n,p}(t_m) \mathbf{n}_{m0}^T \\ \vdots & \cdots & \vdots \\ N_{0,p}(t_m) \mathbf{n}_{mN}^T & \cdots & N_{n,p}(t_m) \mathbf{n}_{mN}^T \end{pmatrix}, \tag{4.8}$$

$$\mathbf{b} = (\mathbf{b}_0, \dots, \mathbf{b}_m)^T, \tag{4.9}$$

$$\mathbf{b}_i = (\mathbf{q}_{i0}^T \mathbf{n}_{i0}, \dots, \mathbf{q}_{iN}^T \mathbf{n}_{iN}), \quad i = 0, \dots, m, \tag{4.10}$$

\mathbf{A} is the coefficient matrix and \mathbf{b} is a constant term. So we get:

$$\min E = \min(\|\mathbf{A} \mathbf{x} - \mathbf{b}\|_1 + \mu \mathbf{x}^T \mathbf{K} \mathbf{x}). \tag{4.11}$$

This is an optimization problem defined by the sum of an L1-norm term and a quadratic form term. The solution of the optimization problem is considered as the control points of the repaired model’s ideal feature line. There are some convenient tools to solve the convex optimization problem, such as CVX, a Matlab-based modeling system for convex optimization, and so on.

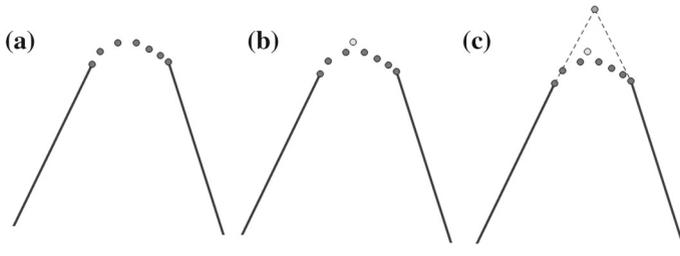


Fig. 4 Sharp feature point recovery based on energy optimization in 2-dimensional space. The *dark points* denote points added to repair the model. The *light point* is the modified point sharpened by the sharpener [2]. The *new point* is the solution of our energy optimization model

In particular, in 2-dimensional circumstances, the feature line degenerates to a feature point. The mode of the second derivative of the point is zero and therefore, energy E can be expressed as follows:

$$E = \sum_{j=0}^N |(\mathbf{P} - \mathbf{q}_j)^T \mathbf{n}_j|,$$

where $\{\mathbf{q}_j\}$ are adjacent points of feature point \mathbf{P} and \mathbf{n}_j are corresponding normals. Solve $\mathbf{P} = \arg \min E$ to obtain the correction of the sharp feature point (see Fig. 4).

The specific steps of the algorithm are shown in Algorithm 1.

Algorithm 1 Sharp feature recovery based on energy optimization

- 1: Fill holes and determine the locations of the sharp feature points on the repaired model.
 - 2: For each feature point \mathbf{Q}_i , determine the adjacent points $\{\mathbf{q}_{ij}\}$ and simultaneously, make $\{\mathbf{q}_{ij}\}$ distributed in almost the same amount on every surface (see Fig. 3).
 - 3: Solving $\mathbf{P} = \arg \min E$, \mathbf{P} denotes the control points of the feature line $P(t)$ of the model.
-

5 Sharp Feature Recovery Based on Plane Cluster

In the previous section, we introduced a method that can recover features well, especially when the features can be considered as curves. However, when the features of a model are sharp corner points or scattered points, we can still use the previous method to modify the feature (treat points as special lines); but we prefer another method to address this situation (Fig. 5).

Denote $\{\mathbf{Q}_i\}_{i=0}^m$ as the feature points of the repaired model. For each feature point \mathbf{Q}_i , there is a set of adjacent points $\{\mathbf{q}_{ij}\}_{j=0}^N$. The specific method to find these adjacent points is stated in Sect. 4. Every adjacent point \mathbf{q}_{ij} determines a plane $\Pi_{\mathbf{q}_{ij}}$ by its position and its normal information and every point set $\{\mathbf{q}_{ij}\}$ determines a plane cluster $\{\Pi_{\mathbf{q}_{ij}}\}$ corresponding to every feature point \mathbf{Q}_i on the sharp feature. Seek the point \mathbf{P}_i that is nearest to the plane cluster $\{\Pi_{\mathbf{q}_{ij}}\}$ under the condition of the least square

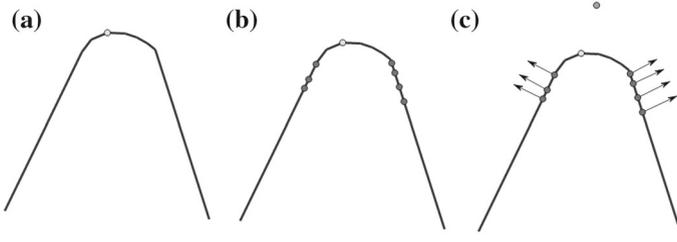


Fig. 5 Sharp feature point recovery based on plane clusters in 2-dimensional space. The *light point* in **a** shows the feature point in the repaired model. The *dark points* in **b** show the adjacent points on the initial model of the *light point*. In **c** planes are established for every adjacent point. The *new point* is the nearest point to the plane cluster by least squares, i.e., the *new point* is the modification of the *light point*

method and treat \mathbf{P}_i as the modification of the initial feature point \mathbf{Q}_i , i.e., find \mathbf{P}_i to minimize 5.1.

$$\min \sum_{j=0}^N |dis(\Pi_{\mathbf{q}_{ij}}, \mathbf{P}_i)|^2. \tag{5.1}$$

The solution of the problem always exists but is not always unique. To avoid singular cases caused by the non-uniqueness of the solution, we can add a term $\mu|\mathbf{P}_i - \mathbf{Q}_i|^2$ to the equation to ensure the solution point is the nearest point to \mathbf{Q}_i , in which, μ is a small constant that can be set manually. Therefore, for each feature point \mathbf{Q}_i , find a point \mathbf{P}_i to minimize 5.2.

$$\min \left(\sum_{j=0}^N |dis(\Pi_{\mathbf{q}_{ij}}, \mathbf{P}_i)|^2 + \mu|\mathbf{P}_i - \mathbf{Q}_i|^2 \right). \tag{5.2}$$

\mathbf{P}_i is a better estimation of the real model sharp feature point than point \mathbf{Q}_i , see Fig. 6. The specific algorithm steps are shown in Algorithm 2.

Algorithm 2 Sharp feature recovery based on plane cluster

- 1: Fill holes and determine the locations of the sharp feature points on the repaired model.
 - 2: For each feature point \mathbf{Q}_i , determine the adjacent points $\{\mathbf{q}_{ij}\}$ and simultaneously, make $\{\mathbf{q}_{ij}\}$ distributed almost the same amount on every surface (see Fig. 3).
 - 3: For each adjacent point \mathbf{q}_{ij} , a plane $\Pi_{\mathbf{q}_{ij}}$ is defined by the position and normal of \mathbf{q}_{ij} .
 - 4: For every plane cluster $\{\Pi_{\mathbf{q}_{ij}}\}$, seek \mathbf{P}_i to solve $\min(\sum_{j=0}^N |dis(\Pi_{\mathbf{q}_{ij}}, \mathbf{P}_i)|^2 + \mu|\mathbf{P}_i - \mathbf{Q}_i|^2)$.
-

We can use the method to cope with the circumstance in which the model features are almost curves: modify every initially picked feature point $\{\mathbf{Q}_i\}$, then a new feature can be defined by approximating the modified points $\{\mathbf{P}_i\}$. The new feature is closer to the intersection of the surfaces and satisfies some smoothness. The new feature is a better choice than the repaired feature. The method first modifies every feature point and then fits them globally. For a single sharp feature point, the correction point is

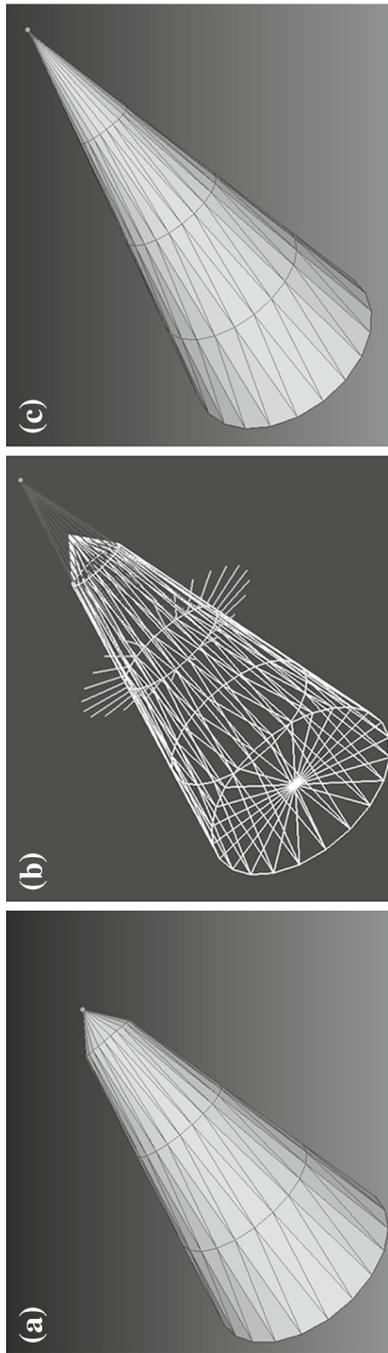


Fig. 6 Sharp feature point recovery based on plane clusters. **a** demonstrates a cone with its apex in wrong position. In **b** the segments around the cone show the normals of adjacent points and we can see that 1-neighbor points are abandoned to get rid of the influence of filling hole points. Establish a plane for each adjacent point and find a point to solve Eq. (5.2). The modified result of the sharp feature point is show in **(c)**

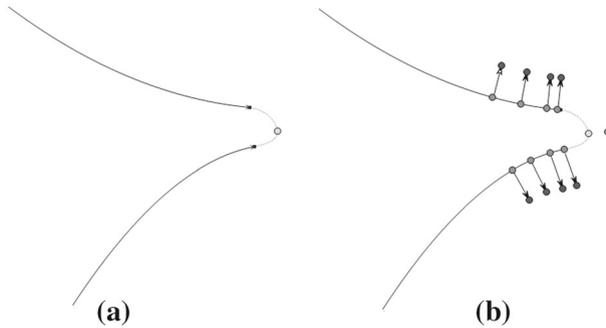


Fig. 7 Curve intersection point in 2-dimensional space. The *thin line* in **a** shows the points added to repair the model, and the *light point* shows the repaired feature point. The *dark points* in **b** denote the adjacent points of the feature point. The *arrows* show the normal direction, and the *new point* is the modified feature point

consistent with the energy optimization method in 2-dimensional space (see Sect. 4). For the 3-dimensional case, the plane cluster approach fits the points after each point is modified whereas the energy optimization method solves the problem globally. Both algorithms can handle sharp feature recovery, but they have their own advantages in handling different conditions. The choice of the proper method depends on the model's features (Fig. 7).

6 Results and Discussion

We have implemented the techniques presented in the previous section and tested them on a large number of models. We report here on some instances of sharp feature recovery. In all of the examples tested, when the feature of the original repaired model is not far from the theoretical position, most of the sharp features can be successfully recovered. Compared with the method proposed in [2,28], our approaches are more efficient and work better because our results not only recover the sharp features but also keep them smooth with the model globally.

Figure 8 shows an instance which is a cube with its edges lost. We can see that the repair result is not good with handling its features. It is critical to find the adjacent points in both methods proposed above. We should try to pick the same amount from every surface, as shown in Fig. 8c. We abandon the 1-neighborhood points to avoid the wrong normal information. Figure 8d, e show the comparison of the initial model and our result.

Figure 9a indicates the initial breakage model; Fig. 9b is the repaired result. The feature of the repaired model is not good. Ju's repair and sharp feature reproduction method is tested in Fig. 9c. We use the PloyMender-dc version 1.7.1 provided by the author. The octree depth and scale we used here are 6 and 0.9. Otherwise, we tested the same model by the method used by Attene et al. in [1]. And Fig. 9d presents the result of feature recovery by processing twice by the edge sharpener [2]. The threshold angle and threshold length we used here are 25 and 200. The results are

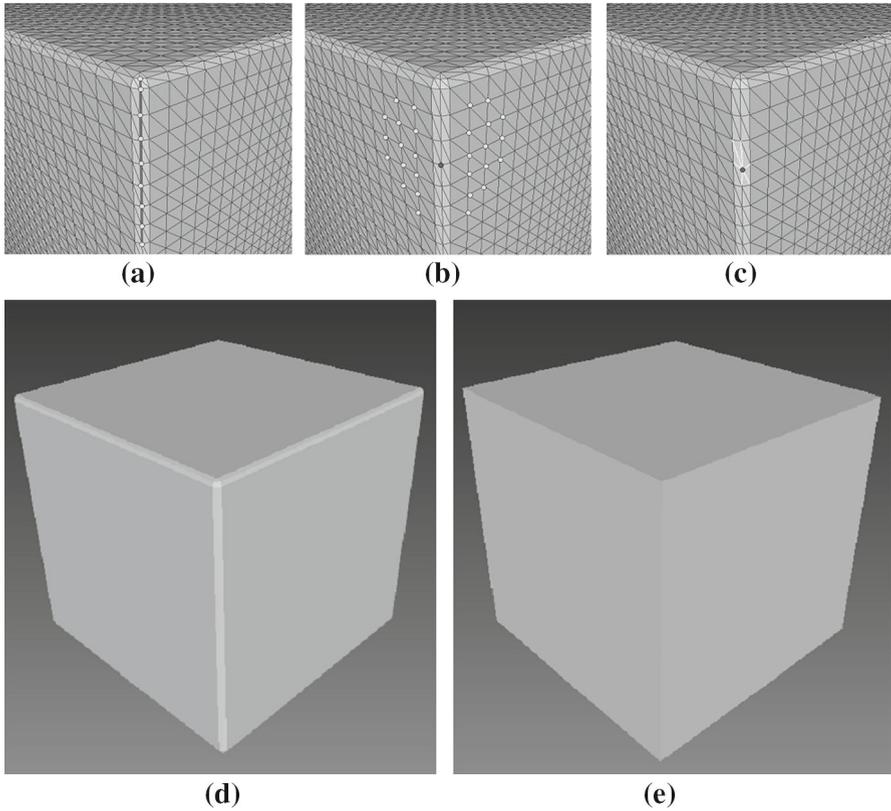


Fig. 8 Sharp feature recovery of a cube lost its edges. The *light points* in **a** are added points in *filling holes* process, and the *dark curve* shows the features in the repaired cube. In **b** the *light points* show the adjacent points of a feature point (shown in *dark*). Moreover, added points and 1-neighbor points are abandoned when choosing adjacent points here. The modified *dark* feature point is shown in **c**. The comparison between the *filling holes* model and our sharp feature recovery result is demonstrated in **(d)** and **(e)**

not satisfactory. In Fig. 9e we can see the repaired feature (red points) and modified feature by our sharp feature recovery based on the energy optimization algorithm (green lines). The constant μ used here is 0.1 and 6 adjacent points for each feature point, i.e., $N = 6$. We used 3rd-degree B-spline curve with 10 control points to fit feature line and chose 212 feature points from the repaired model. Figure 9f shows our result (Fig. 10).

Figure 11 shows the similar recovery process of the breakage casting model with $\mu = 0.3$ and $N = 8$ (both Algorithms 1 and 2). 143 feature points were chosen from the repaired model and the feature line was approximated by 2nd-degree B-spline curve with 5 control points. Figure 11a–c are the initial raw model and the repaired model [22]. Ju’s and Attene’s results are presented in Fig. 11d, e. Compared to their results, our sharp feature recovery results are better, see Fig. 11h, i. From Fig. 11f, g we can clearly see that the results of sharp feature recovery based on Algorithms 1 and 2 have some differences. The shape of feature lines are similar and the remesh result both keep the sharp feature well. Both the missing regions in Figs. 9 and 11 are

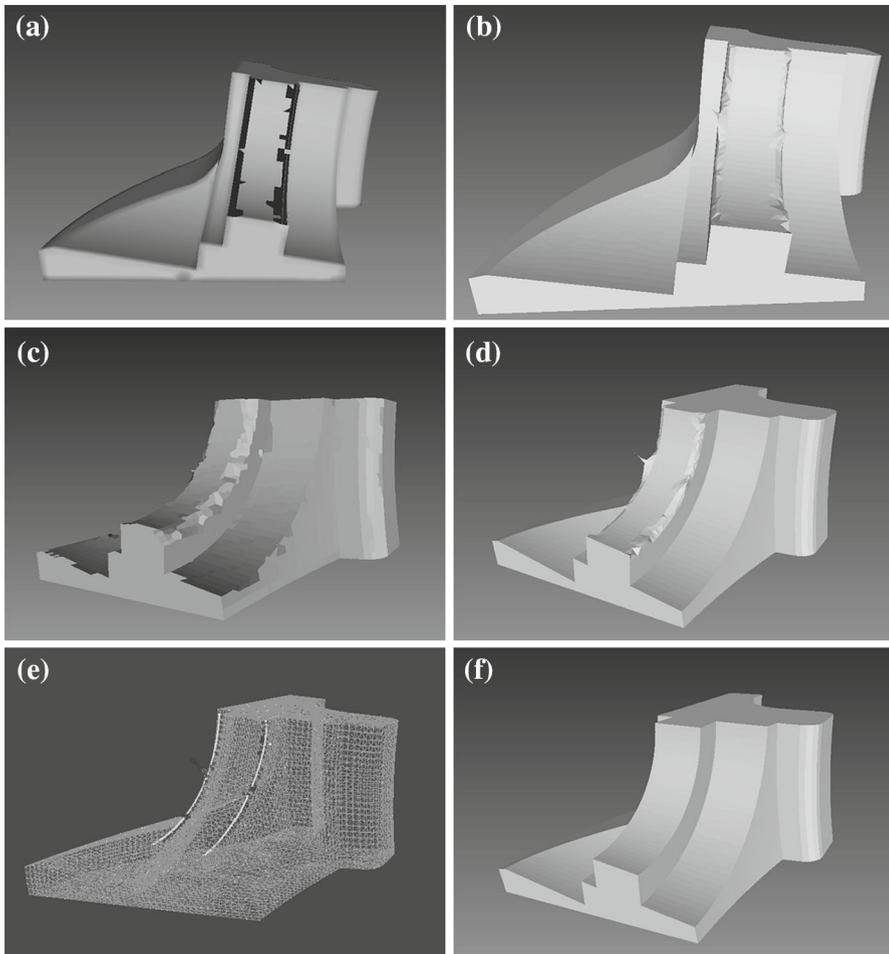


Fig. 9 Sharp feature recovery of a fandisk model. **a** indicates the initial breakage model, and **b** is the result of *filling holes* by the method in [22]. **c**, **d**, respectively, show the result of the feature recovery results by [18] and [1]. The *dark points* in **e** are the detected feature points, and the *light lines* are our solution of the ideal feature. **f** presents our result

removed manually to test the algorithms. The following example comes from scanning process.

When we attempted to reconstruct a 3-dimensional entity model by the scanning-reconstruction method, a problem occurred, as mentioned in previous section. Figure 12 presents our method to recover sharp feature in contrast to the method introduced in [2]. Figure 12a, b demonstrate the problem we mentioned in Sect. 1. Figure 12a, c and d show the raw scanning model, the features, and the repaired model, and simultaneously, they indicate that the sharpener method can recover sharp features to some extent but the result is not satisfactory, especially with respect to the global smoothness of the feature. Our processing of sharp feature recovery result is demonstrated in

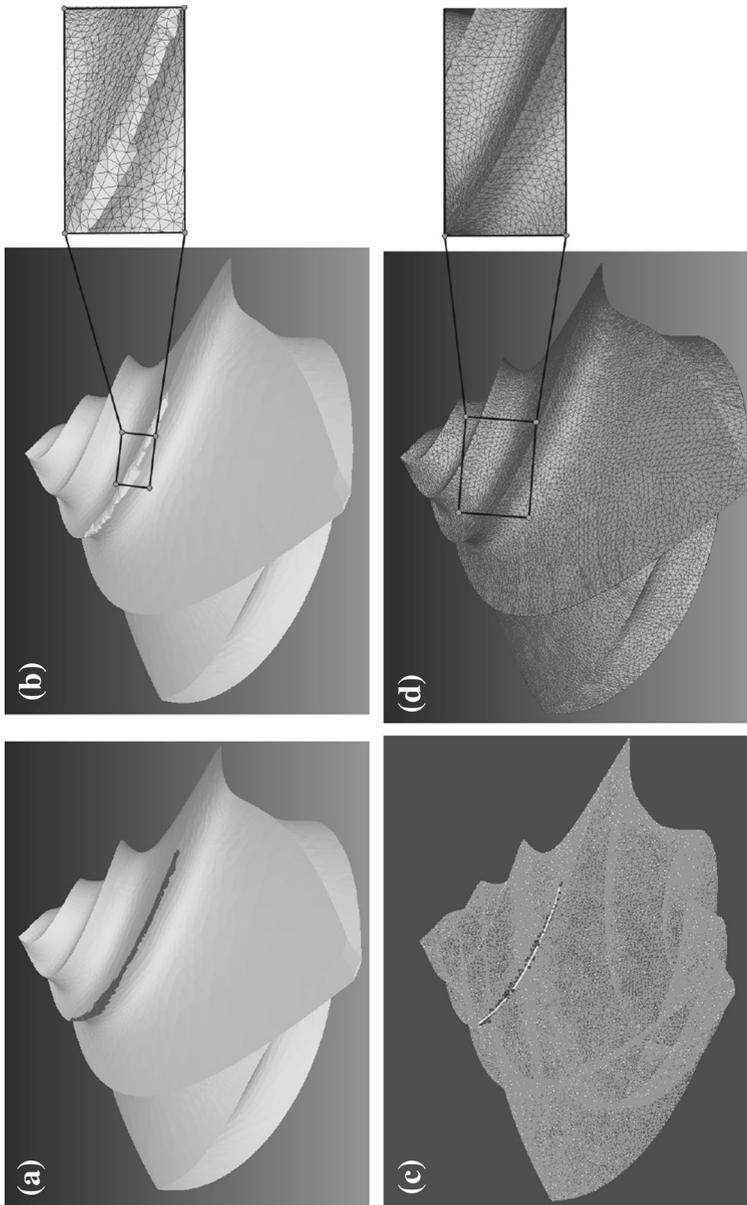


Fig. 10 Sharp feature recovery of Octa-flower model. **a** indicates the initial breakage model and the corresponding *filling hole* result is shown in **(b)**. The *dark points* in **c** are repair feature points. The recovery sharp feature *line* by Algorithm 2 ($\mu = 0.2, N = 6$) is *light*. **d** presents our remesh result

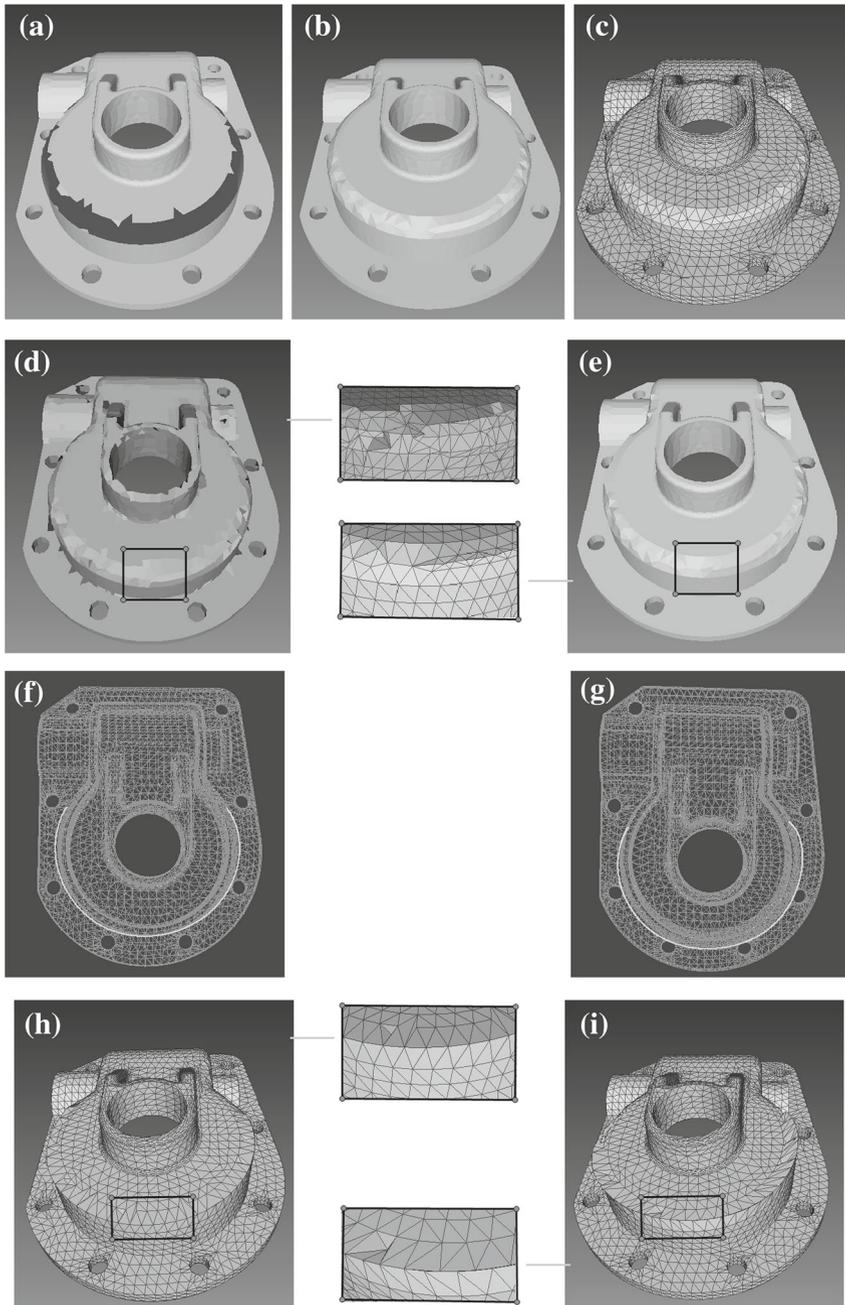


Fig. 11 Sharp feature recovery of a casting model. **a**, **b**, and **c** are the initial raw model and the repaired model. Ju's and Attene's results are presented in **(d)** and **(e)**. **f**, **g** show the sharp feature recovery result by Algorithms 1 and 2 where feature points are *dark* and sharp feature *lines* are *light*. Our remesh results are demonstrated in **(h)** and **(i)**

Fig. 12f, g. For the purpose of 3-dimensional reconstruction, we approximated the surfaces to obtain the NURBS expression of the model and remeshed the model Fig. 12g. The detailed feature condition is clear in the enlarged portions. From the result, it is easy to understand that our method not only restores the sharp feature information more effectively but also keeps features smooth along with the initial model. The parameters used in this situation are $\mu = 0.1$, $p = 3$, $m = 1183$ and $N = 12$. It is a real model and problem in our practice of reconstruction rather than manual data, and the result proves that our method can be used in practical 3-dimensional entity reconstruction. The parameters used here should not be too large or too small. As

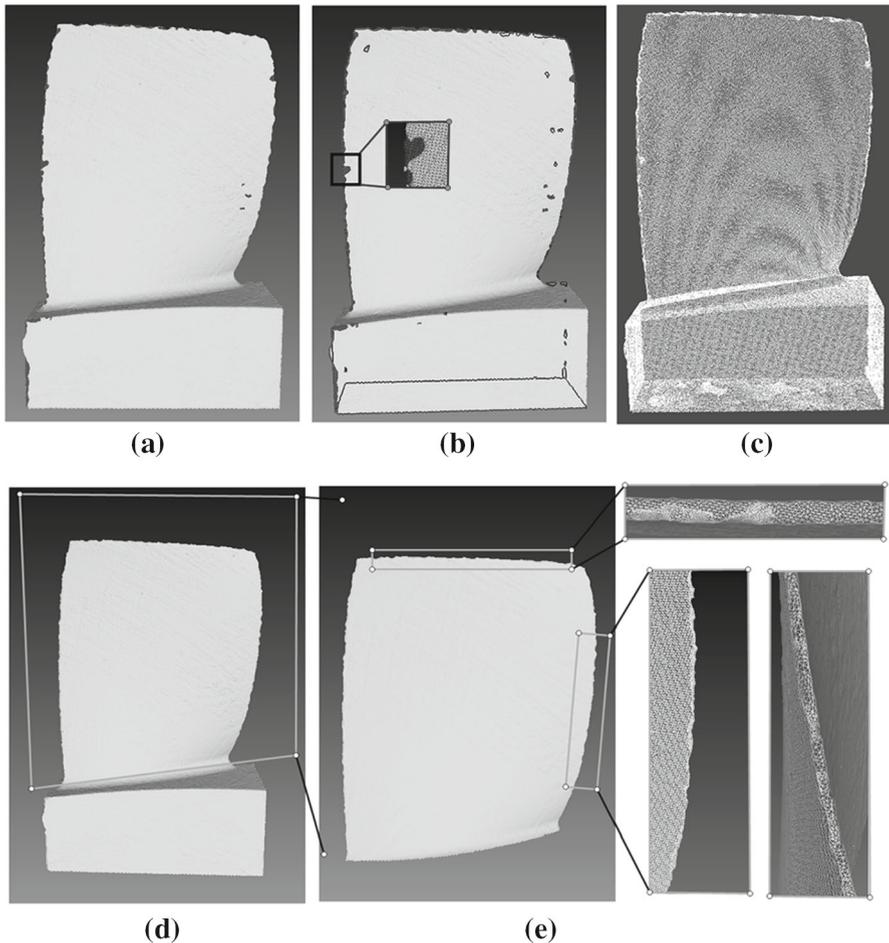


Fig. 12 Sharp feature recovery of an entity model. **a, b** are the initial raw scanned models. **c–e** show the repaired result and the result has been denoised. The *dark points* in **c** show the feature points of the repaired model. After sharp feature recovery, the blade part of the model was remeshed by taking the sharp feature curve as the boundary of the model. The recovered sharp feature by Algorithm 1 and Algorithm 2 are shown in **(f)** and **(h)**. **g** and **i** show the remesh result

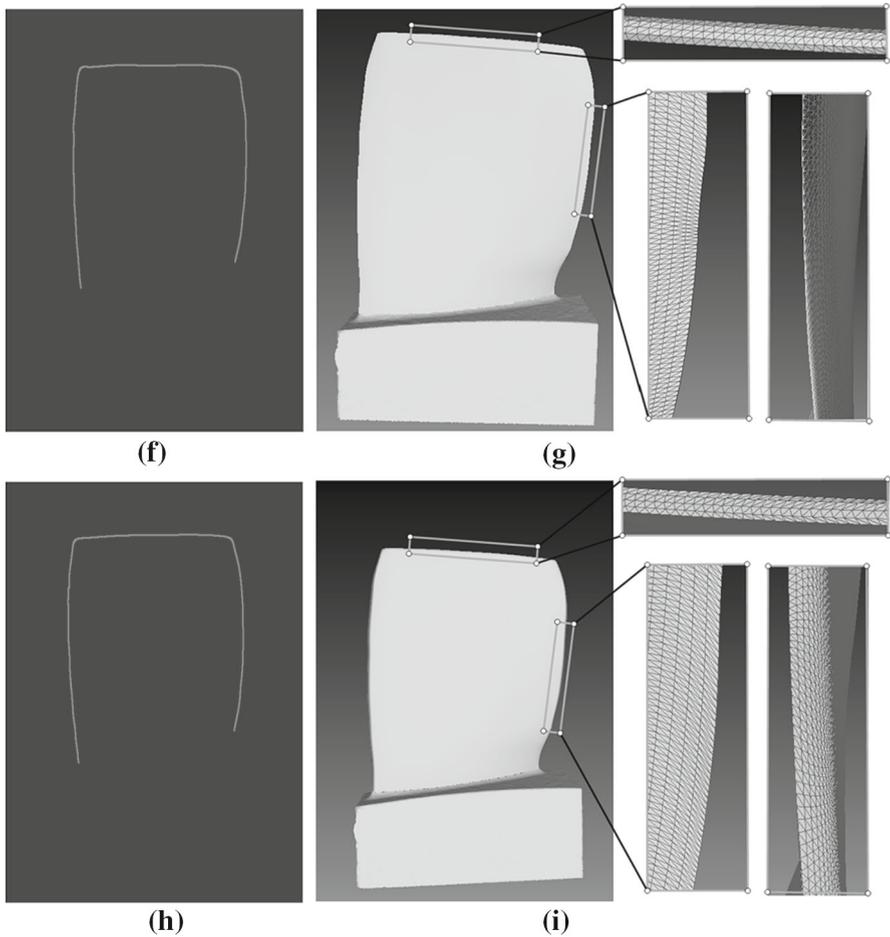


Fig. 12 continued

stated before, μ in Eq. 4.11 is a constant coefficient of smooth term. Under normal circumstances, the larger μ is, the smoother feature lines become. At the same time, feature lines would shrink. In 5.2, μ is the constant coefficient. If its value is too small, it may cause the multiple resolution of the problem. On the other hand, if μ is too large, the result will be too close to the initial feature points. In our experiments, we usually use 0.1–0.5 as the value of μ . The appropriate value of N depends on the scale of the model and the area around the feature line. If we choose too many adjacent points, some irrelevant points could be used to compute feature line and if the adjacent points are not enough, the position of the feature line would be imprecise. Usually, we take 6–10 as the value of N for small models and 12–20 for big models.

Table 1 reports the time consumed for sharp feature recovery and model remesh in the paper. The experiments were performed on a consumer-level PC with a 2.6

Table 1 Performance results from repairing various models on a consumer-level PC with a 2.6 GHz CPU and 2 GB memory

Model	Input vertices	Input triangles	Feature recovery time (s)	Remesh time (s)	Output vertices	Output triangles
Cone	98	192	0.57	0	98	192
Cube	2096	5080	2.43	0.572	2096	5080
Fandisk	6556	13099	3.169	3.320	6574	13072
Octa-flower	39643	79140	3.969	5.550	39755	79326
Casting	4981	9880	2.894	2.147	5020	10072
Entity	203654	407234	6.232	9.654	118590	237412

GHz CPU and 2 GB memory. Compared with the method approached in [28], our algorithms are more efficient.

7 Conclusion

In this paper, we have proposed two methods for processing sharp feature recovery to cope with the lack of existing methods. Our methods are simple, automatic and efficient, and more importantly, they depend little on the result of filling holes. Based on these conditions, even the initial result of repairing is not good, as long as the adjacent points of the initial repaired feature can be picked up correctly, our methods can work well. Taking the importance and time-consuming characteristic of finding adjacent points into account, we can use the ANN library and KD-Tree to improve our operational efficiency and accuracy. Besides, as mentioned in Sect. 4, CVX is helpful for improving the efficiency.

Our methods also have some limitations. The algorithms presented here are sensitive to noise because both curve fitting and corner refinement rely on normal information. Therefore, we should denoise the model first. Moreover, as stated in Sect. 6 the parameters of the algorithms will influence the result in different aspects. Users can adjust the parameters to make a satisfactory result.

In the process of model reconstruction or reverse engineering, for the breakage scanned model, we can use the following method to repair the model and maintain the sharp feature information at the same time: first, fill holes to repair initial model. Second, use one of the algorithms proposed in this paper to modify the sharp features and delete the filling hole points and the repaired features. Finally, approximate the initial scanned data and the restored sharp features. Then we can get a model that not only approximates the entity model well but also maintains the model's sharp features. Because the recovery process should be executive interactively, the accuracy of the sharp feature positions is highly improved.

Acknowledgments The authors are supported by a NKBRPC(2011CB302400), the National Natural Science Foundation of China (11171322 and 11371341), and the 111 Project (No. b07033).

References

1. Attene, M., Falcidieno, B.: Remesh: an interactive environment to edit and repair triangle meshes. In: IEEE International Conference on Shape Modeling and Applications, 2006 (SMI 2006), pp. 41–41 (2006)
2. Attene, M., Falcidieno, B., Rossignac, J., Spagnuolo, M.: Edge-sharpener: recovering sharp features in triangulations of non-adaptively re-meshed surfaces. In: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, pp. 62–69. Eurographics Association, June (2003)
3. Attene, M., Falcidieno, B., Rossignac, J., Spagnuolo, M.: Sharpen&bend: recovering curved sharp edges in triangle meshes produced by feature-insensitive sampling. *IEEE Trans. Vis. Comput. Graph.* **11**(2), 181–192 (2005)
4. Avron, H., Sharf, A., Greif, C., Cohen-Or, D.: L1-sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph. (TOG)* **29**(5), 135 (2010)
5. Barequet, G., Kumar, S.: Repairing cad models. In: IEEE Visualization '97, pp. 363–370 (1997)
6. Barequet, G., Sharir, M.: Filling gaps in the boundary of a polyhedron. *Comput. Aided Geom. Des.* **12**(2), 207–229 (1995)
7. Biermann, H., Martin, I.M., Zorin, D., Bernardini, F.: Sharp features on multiresolution subdivision surfaces. *Graph. Models* **64**(2), 61–77 (2002)
8. Chen, C.-Y., Cheng, K.-Y., Liao, H.M.: A sharpness dependent approach to 3d polygon mesh hole filling. In: Proceedings of EuroGraphics, pp. 13–16 (2005)
9. Daniels, J., Ha, L.K., Ochotta, T., Silva, C.T.: Robust smooth feature extraction from point clouds. In: IEEE International Conference on Shape Modeling and Applications, 2007 (SMI'07), pp. 123–136 (2007)
10. Davis, J., Marschner, S.R., Garr, M., Levoy, M.: Filling holes in complex surfaces using volumetric diffusion. In: IEEE Proceedings, First International Symposium on 3D Data Processing Visualization and Transmission, 2002, pp. 428–441 (2002)
11. Desbrun, M., Meyer, M., Schröder, P., Barr, A.H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In: Proceedings of SIGGRAPH 99, Computer Graphics Proceedings, Annual Conference Series, pp. 317–324, August (1999)
12. Fleishman, S., Cohen-Or, D., Silva, C.T.: Robust moving least-squares fitting with sharp features. *ACM Trans. Graph.* **24**(3), 544–552 (2005)
13. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. *ACM Trans. Graph.* **22**(3), 950–953 (2003)
14. Hoppe, H.: Progressive meshes. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 99–108. ACM (1996)
15. Huang, H., Wu, S., Gong, M., Cohen-Or, D., Ascher, U., Zhang, H.R.: Edge-aware point set resampling. *ACM Trans. Graph. (TOG)* **32**(1), 9 (2013)
16. Hubeli, A., Gross, M.: Multiresolution feature extraction for unstructured meshes. *IEEE Vis.* **2001**, 287–294 (2001)
17. Jones, T.R., Durand, F., Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. *ACM Trans. Graph.* **22**(3), 943–949 (2003)
18. Ju, T.: Robust repair of polygonal models. *ACM Trans. Graph.* **23**(3), 888–895 (2004)
19. Ju, T., Losasso, F., Schaefer, S., Warren, J.: Dual contouring of hermite data. *ACM Trans. Graph.* **21**(3), 339–346 (2002)
20. Kobbelt, L.P., Botsch, M., Schwanecke, U., Seidel, H.-P.: Feature sensitive surface extraction from volume data. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, Computer Graphics Proceedings, Annual Conference Series, pp. 57–66. ACM August (2001)
21. Lai, Y.-K., Zhou, Q.-Y., Hu, S.-M., Wallner, J., Pottmann, H.: Robust feature classification and editing. *IEEE Trans. Vis. Comput. Graph.* **13**(1), 34–45 (2007)
22. Liepa, P.: Filling holes in meshes. In: Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 200–205. Eurographics Association, June (2003)
23. Lipman, Y., Cohen-Or, D., Levin, D.: Data-dependent mls for faithful surface approximation. In: Fifth Eurographics Symposium on Geometry Processing, pp. 59–68, July (2007)
24. Nooruddin, F.S., Turk, G.: Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.* **9**(2), 191–205 (2003)

25. Sharf, A., Alexa, M., Cohen-Or, D.: Context-based surface completion. *ACM Trans. Graph.* **23**(3), 878–887 (2004)
26. Sun, X., Rosin, P., Martin, R., Langbein, F.: Fast and effective feature-preserving mesh denoising. *IEEE Trans. Vis. Comput. Graph.* **13**(5), 925–938 (2007)
27. Taubin, G.: A signal processing approach to fair surface design. In: *Proceedings of SIGGRAPH 95, Computer Graphics Proceedings, Annual Conference Series*, pp. 351–358, August (1995)
28. Wang, C.C.: Bilateral recovering of sharp edges on feature-insensitive sampled meshes. *IEEE Trans. Vis. Comput. Graph.* **12**(4), 629–639 (2006)
29. Wang, R., Yang, Z., Liu, L., Deng, J., Chen, F.: Decoupling noises and features via weighted l1-analysis compressed sensing. *ACM Trans. Graph.* **33**(2), 1–12 (2014). Article 18
30. Wang, X., Liu, X., Lu, L., Li, B., Cao, J., Yin, B., Shi, X.: Automatic hole-filling of cad models with feature-preserving. *Comput. Graph.* **36**(2), 101–110 (2012)
31. Watanabe, K., Belyaev, A.G.: Detection of salient curvature features on polygonal surfaces. *Comput. Graph. Forum* **20**(3), 385–392 (2001)
32. Yagou, H., Ohtake, Y., Belyaev, A.: Mesh smoothing via mean and median filtering applied to face normals. In: *IEEE Proceedings of the Geometric Modeling and Processing, 2002*, pp. 124–131 (2002)